CS 188: Artificial Intelligence

Bayes Nets: Exact Inference

Instructors: Angela Liu and Yanlai Yang
University of California, Berkeley
(Slides adapted from Stuart Russell and Dawn Song)
A Bayes net is an efficient encoding of a probabilistic model of a domain

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over \( X \), one for each combination of parents’ values

\[
P(X|a_1 \ldots a_n)
\]

Bayes’ nets implicitly encode joint distributions

- As a product of local conditional distributions
- To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

\[
P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i))
\]
Size of a Bayes Net

- How big is a joint distribution over $N$ Boolean variables?
  
  $2^N$

- How big is an $N$-node net if nodes have up to $k$ parents?
  
  $O(N \times 2^{k+1})$

- Both give you the power to calculate
  
  $P(X_1, X_2, \ldots, X_n)$

- BNs: Huge space savings!

- Also easier to elicit local CPTs

- Also faster to answer queries (this lecture!)
Causality?

- BNs need not actually be causal
  - Sometimes no causal net exists over the domain (especially if variables are missing)
  - E.g. consider the variables Traffic and Rain

\[
P(R)
\begin{array}{c|c}
+r & 1/4 \\
-r & 3/4 \\
\end{array}
\]

\[
P(T)
\begin{array}{c|c}
+t & 9/16 \\
-t & 7/16 \\
\end{array}
\]

\[
P(T|R)
\begin{array}{c|c|c}
+r & +t & 3/4 \\
-r & +t & 1/2 \\
+r & -t & 1/4 \\
-r & -t & 1/2 \\
\end{array}
\]

\[
P(R|T)
\begin{array}{c|c|c}
+t & +r & 1/3 \\
+t & -r & 2/3 \\
-t & +r & 1/7 \\
-t & -r & 6/7 \\
\end{array}
\]
Example: Traffic

- Causal direction

\[
P(R) = \begin{array}{|c|c|} 
+ r & 1/4 \\
- r & 3/4 \\
\end{array}
\]

\[
P(T | R) = \begin{array}{|c|c|c|} 
+ r & + t & 3/4 \\
+ r & - t & 1/4 \\
- r & + t & 1/2 \\
- r & - t & 1/2 \\
\end{array}
\]

\[
P(T, R) = \begin{array}{|c|c|} 
+ r & + t & 3/16 \\
+ r & - t & 1/16 \\
- r & + t & 6/16 \\
- r & - t & 6/16 \\
\end{array}
\]
Example: Reverse Traffic

- Reverse causality?

\[
P(T) \begin{array}{c|c}
+ t & 9/16 \\
- t & 7/16 \\
\end{array}
\]

\[
P(R | T) \begin{array}{c|c|c}
+ t & + r & 1/3 \\
- t & + r & 1/7 \\
- r & 2/3 & \\
\end{array}
\]

\[
P(T, R) \begin{array}{c|c|c}
+ r & + t & 3/16 \\
+ r & - t & 1/16 \\
- r & + t & 6/16 \\
- r & - t & 6/16 \\
\end{array}
\]
Causality?

- When Bayes’ nets reflect the true causal patterns:
  - Often simpler (nodes have fewer parents)
  - Often easier to think about
  - Often easier to estimate probabilities from data

- BNs need not actually be causal
  - Sometimes no causal net exists over the domain (especially if variables are missing)
  - E.g. consider the variables Traffic and Drips
  - End up with arrows that reflect correlation, not causation

- What do the arrows really mean?
  - Topology may happen to encode causal structure
  - Topology really encodes conditional independence
    \[ P(x_i|x_1, \ldots x_{i-1}) = P(x_i|\text{parents}(X_i)) \]
Reminder of inference by enumeration:
- Any probability of interest can be computed by summing entries from the joint distribution: \( P(Q | e) = \alpha \sum_h P(Q, h, e) \)
- Entries from the joint distribution can be obtained from a BN by multiplying the corresponding conditional probabilities

\[
P(B | j, m) = \alpha \sum_{e,a} P(B, e, a, j, m)
\]

\[
= \alpha \sum_{e,a} P(B) P(e) P(a|B,e) P(j|a) P(m|a)
\]

Problem: sums of \textit{exponentially many} products!
Variable elimination: The basic ideas

- Move summations inwards as far as possible
  - \[ P(B \mid j, m) = \alpha \sum_{e,a} P(B) P(e) P(a \mid B,e) P(j \mid a) P(m \mid a) \]
  - \[ = \alpha P(B) \sum_e P(e) \sum_a P(a \mid B,e) P(j \mid a) P(m \mid a) \]

- Do the calculation from the inside out
  - I.e., sum over \( a \) first, then sum over \( e \)
  - Challenge: \( P(a \mid B,e) \) isn’t a single number, it’s a table of different numbers (depending on the values of \( B \) and \( e \))
  - Solution: use arrays of numbers with appropriate operations on them; these are called factors
Joint distribution: $P(X,Y)$
- Entries $P(x,y)$ for all $x$, $y$
- $|X| \times |Y|$ matrix
- Sums to 1

Projected joint: $P(x,Y)$
- A slice of the joint distribution
- Entries $P(x,y)$ for one $x$, all $y$
- $|Y|$-element vector
- Sums to $P(x)$

$$P(A,J)$$

<table>
<thead>
<tr>
<th>A \ J</th>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>0.09</td>
<td>0.01</td>
</tr>
<tr>
<td>false</td>
<td>0.045</td>
<td>0.855</td>
</tr>
</tbody>
</table>

$$P(a,J) = P_a(J)$$

<table>
<thead>
<tr>
<th>A \ J</th>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>0.09</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Number of variables (capitals) = dimensionality of the table
Single conditional: $P(Y \mid x)$
- Entries $P(y \mid x)$ for fixed $x$, all $y$
- Sums to 1

Family of conditionals: $P(X \mid Y)$
- Multiple conditionals
- Entries $P(x \mid y)$ for all $x$, $y$
- Sums to $|Y|$
Operation 1: Pointwise product

- First basic operation: **pointwise product** of factors (*not* matrix multiply!)
  - New factor has union of variables of the two original factors
  - Each entry is the product of the corresponding entries from the original factors

- Example: $P(J|A) \times P(A) = P(A,J)$

| $P(A)$          | $P(J|A)$                   | $P(A,J)$                   |
|----------------|----------------------------|----------------------------|
| true           | 0.1                        | true 0.9, false 0.1        |
| false          | 0.9                        | true 0.05, false 0.95      |
| $\times$       |                            | $=$                         |
|                 |                            | true 0.09, false 0.01      |
|                 |                            | true 0.045, false 0.855    |
Example: Making larger factors

- Example: $P(A,J) \times P(M|A) = P(A,J,M)$
- Factor blowup can make VE very expensive!

\[
P(A,J) \quad \begin{array}{c|c|c} A \setminus J & \text{true} & \text{false} \\
\hline \text{true} & 0.09 & 0.01 \\
\text{false} & 0.045 & 0.855 \\
\end{array}
\quad \times \quad P(M|A) \quad \begin{array}{c|c|c} A \setminus M & \text{true} & \text{false} \\
\hline \text{true} & 0.7 & 0.3 \\
\text{false} & 0.01 & 0.99 \\
\end{array}
\quad = \quad P(A,J,M) \quad \begin{array}{c|c|c} J \setminus M & \text{true} & \text{false} \\
\hline \text{true} & \text{false} & \text{false} \\
\text{false} & \text{false} & 0.003 \\
\end{array}
\quad A=\text{false} \quad A=\text{true}
Operation 2: Summing out a variable

- Second basic operation: *summing out* (or eliminating) a variable from a factor
  - Shrinks a factor to a smaller one
- Example: \( \sum_j P(A,J) = P(A,j) + P(A,\neg j) = P(A) \)

<table>
<thead>
<tr>
<th>( A \setminus J )</th>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>0.09</td>
<td>0.01</td>
</tr>
<tr>
<td>false</td>
<td>0.045</td>
<td>0.855</td>
</tr>
</tbody>
</table>

Sum out \( J \)

<table>
<thead>
<tr>
<th>( P(A) )</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>0.1</td>
</tr>
<tr>
<td>false</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Variable Elimination
Variable Elimination

- Query: \( P(Q|E_1=e_1,..., E_k=e_k) \)

- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable \( H_j \)
  - Eliminate (sum out) \( H_j \) from the product of all factors mentioning \( H_j \)

- Join all remaining factors and normalize
Example

Query $P(B \mid j,m)$

Choose A

$P(A \mid B,E)\; \times \; P(j \mid A)\; \times \; P(m \mid A) \quad \sum \quad P(j,m \mid B,E)$

$P(B)\; P(E)\; P(j,m \mid B,E)$
Example

Choose E

\[
P(E) 
\times 
\sum 
P(j,m|B,E) 
= 
P(j,m|B)
\]

Finish with B

\[
P(B) 
\times 
P(j,m,B) 
\xrightarrow{\text{Normalize}} 
P(B | j,m)
\]
Summing out from a product of factors

- Project the factors each way first, then sum the products
- Example: \( \sum_a P(a \mid B, e) \times P(j \mid a) \times P(m \mid a) \)
  
  \[ = P(a \mid B, e) \times P(j \mid a) \times P(m \mid a) + \]

  \[ P(\neg a \mid B, e) \times P(j \mid \neg a) \times P(m \mid \neg a) \]
Order matters

- **Order the terms Z, A, B, C, D**
  
  \[ P(D) = \alpha \sum_{a,b,c} P(z) P(a|z) P(b|z) P(c|z) P(D|z) \]
  
  \[ = \alpha \sum_z P(z) \sum_a P(a|z) \sum_b P(b|z) \sum_c P(c|z) P(D|z) \]
  
  Largest factor has 2 variables (D,Z)

- **Order the terms A, B, C, D, Z**
  
  \[ P(D) = \alpha \sum_{a,b,c,z} P(a|z) P(b|z) P(c|z) P(D|z) P(z) \]
  
  \[ = \alpha \sum_a \sum_b \sum_c \sum_z P(a|z) P(b|z) P(c|z) P(D|z) P(z) \]
  
  Largest factor has 4 variables (A,B,C,D)
  
  In general, with \( n \) leaves, factor of size \( 2^n \)
The computational and space complexity of variable elimination is determined by the largest factor (and it’s space that kills you).

The elimination ordering can greatly affect the size of the largest factor.
- E.g., previous slide’s example $2^n$ vs. 2

Does there always exist an ordering that only results in small factors?
- No!
Variables: $W, X, Y, Z$

CNF clauses:
1. $C_1 = W \lor X \lor Y$
2. $C_2 = Y \lor Z \lor \neg W$
3. $C_3 = X \lor Y \lor \neg Z$

Sentence $S = C_1 \land C_2 \land C_3$

$P(S) > 0$ iff $S$ is satisfiable

$\Rightarrow \text{NP-hard}$

$P(S) = K \times 0.5^n$ where $K$ is the number of satisfying assignments for clauses

$\Rightarrow \text{\#P-hard}$
A polytree is a directed graph with no undirected cycles.

For poly-trees the complexity of variable elimination is *linear in the network size* if you eliminate from the leave towards the roots.
Bayes Nets

Part I: Representation

Part II: Exact inference

- Enumeration (always exponential complexity)
- Variable elimination (worst-case exponential complexity, often better)
- Inference is NP-hard in general

Part III: Independence

Part IV: Approximate Inference